# BME3180 - Machine Learning in Biomedical Engineering

**The Instructor:**

Dr. Görkem SERBES (E-006)

gserbes@yildiz.edu.tr

https://avesis.yildiz.edu.tr/gserbes/

**Assistant:**

Fatih Ekrem ONAT

*fonat@yildiz.edu.tr*

*avesis.yildiz.edu.tr/fonat*

# Linear Regression with multiple variables

**Multiple features (variables).**

| Size (feet$^2$) | Price ($1000) |
|---|---|
| $x$ | $y$ |
| 2104 | 460 |
| 1416 | 232 |
| 1534 | 315 |
| 852 | 178 |
| … | … |

$$f_{w,b}(x) = wx + b$$

# Linear Regression with multiple variables (cont.)

## Multiple features (variables).

| Size (feet²) $x_1$ | Number of bedrooms $x_2$ | Number of floors $x_3$ | Age of home (years) $x_4$ | Price ($1000) |
|---|---|---|---|---|
| 2104 | 5 | 1 | 45 | 460 |
| 1416 | 3 | 2 | 40 | 232 |
| 1534 | 3 | 2 | 30 | 315 |
| 852 | 2 | 1 | 36 | 178 |
| … | … | … | … | … |

Notation:

$x_j$ = $j^{th}$ feature

$n$ = number of features

$\vec{x}^{(i)}$ = features of $i^{th}$ training example

$x_j^{(i)}$ = value of feature $j$ in $i^{th}$ training example

# Linear Regression with multiple variables (cont.)

## Model:

Previously: $f_{w,b}(x) = wx + b$

$$f_{w,b}(X) = w_1 X_1 + w_2 X_2 + w_3 X_3 + w_4 X_4 + b$$

example

$$f_{w,b}(X) = 0.1 X_1 + 4 X_2 + 10 X_3 + -2 X_4 + 80$$

$X_1$: size; $X_2$: #bedrooms; $X_3$: #floors; $X_4$: years; 80: base price

$$f_{w,b}(\mathbf{x}) = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$$

# Linear Regression with multiple variables (cont.)

$$f_{\vec{w},b}(\vec{x}) = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$$

$\vec{w} = [w_1 \; w_2 \; w_3 \ldots w_n]$ parameters of the model

$b$ is a number

vector $\vec{x} = [x_1 \; x_2 \; x_3 \ldots x_n]$

$$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b = w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_nx_n + b$$

dot product

multiple linear regression

# Vectorization

## Parameters and features

$\vec{w} = [w_1 \quad w_2 \quad w_3]$

$b$ is a number

$\vec{x} = [x_1 \quad x_2 \quad x_3]$

linear algebra: count from 1

NumPy

w[0] w[1] w[2]

```
w = np.array([1.0,2.5,-3.3])
b = 4
x = np.array([10,20,30])
```

x[0] x[1] x[2]

code: count from 0

## Without vectorization

$f_{\vec{w},b}(\vec{x}) = w_1x_1 + w_2x_2 + w_3x_3 + b$

```
f = w[0] * x[0] +
    w[1] * x[1] +
    w[2] * x[2] + b
```

## Without vectorization

$f_{\vec{w},b}(\vec{x}) = \left(\sum_{j=1}^{n} w_j x_j\right) + b$

$\sum_{j=1}^{n} \rightarrow j = 1 \ldots n$ 1, 2, 3

range(0,n) → j = 0...n-1

```
f = 0          range(n)
for j in range(0,n):
    f = f + w[j] * x[j]
f = f + b
```

## Vectorization

$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$

```
f = np.dot(w,x) + b
```

# Vectorization (cont.)

Without vectorization

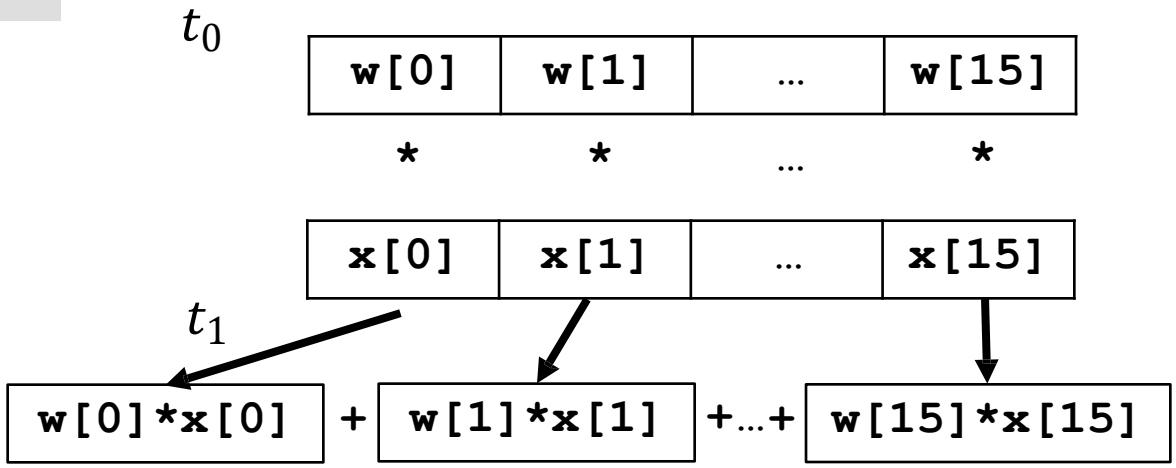
```
for j in range(0,16):
    f = f + w[j] * x[j]
```

$t_0$
```
f + w[0] * x[0]
```
$t_1$
```
f + w[1] * x[1]
```
…

$t_{15}$
```
f + w[15] * x[15]
```

Vectorization

```
np.dot(w,x)
```



# Gradient descent for multiple variables (cont.)

| | Previous notation | Vector notation |
|---|---|---|
| Parameters | $w_1, \cdots, w_n$<br>$b$ | $\vec{\mathrm{w}} = [w_1 \quad \cdots \quad w_n]$<br>$b$ |
| Model | $f_{\vec{\mathrm{w}},b}(\vec{\mathrm{x}}) = w_1 x_1 + \cdots + w_n x_n + b$ | $f_{\vec{\mathrm{w}},b}(\vec{\mathrm{x}}) = \vec{\mathrm{w}} \cdot \vec{\mathrm{x}} + b$ |
| Cost function | $J(w_1, \cdots, w_n, b)$ | $J(\vec{\mathrm{w}}, b)$ |

Gradient descent

repeat {

$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(w_1, \cdots, w_n, b)$

$b = b - \alpha \frac{\partial}{\partial b} J(w_1, \cdots, w_n, b)$

}

repeat {

$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{\mathrm{w}}, b)$

$b = b - \alpha \frac{\partial}{\partial b} J(\vec{\mathrm{w}}, b)$

}

# Gradient descent for multiple variables (cont.)

One feature

repeat {

$$w = w - \alpha \frac{1}{m}\sum_{i=1}^{m}\left(f_{w,b}\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)}$$

$$\frac{\partial}{\partial w}J(w,b)$$

$$b = b - \alpha \frac{1}{m}\sum_{i=1}^{m}\left(f_{w,b}\left(x^{(i)}\right) - y^{(i)}\right)$$

simultaneously update $w$, $b$

}

$n$ features ($n \geq 2$)

repeat {

$$w_1 = w_1 - \alpha \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)}\right)x_1^{(i)}$$

$$\frac{\partial}{\partial w_1}J(\vec{w},b)$$

$\vdots$

$$w_n = w_n - \alpha \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)}\right)x_n^{(i)}$$

$$b = b - \alpha \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)}\right)$$

simultaneously update

$w_j$ (for $j = 1, \cdots, n$) and $b$

}

# Vectorization – Gradient Descent

Gradient descent

$$\vec{w} = (w_1 \quad w_2 \quad \cdots \quad w_{16})$$
$$\vec{d} = (d_1 \quad d_2 \quad \cdots \quad d_{16})$$

```
w = np.array([0.5, 1.3, … 3.4])
d = np.array([0.3, 0.2, … 0.4])
```

compute $w_j = w_j - 0.1 d_j$ for $j = 1 \ldots 16$

Without vectorization

$$w_1 = w_1 - 0.1 d_1$$
$$w_2 = w_2 - 0.1 d_2$$
$$\vdots$$
$$w_{16} = w_{16} - 0.1 d_{16}$$

```
for j in range(0,16):
w[j] = w[j] - 0.1 * d[j]
```

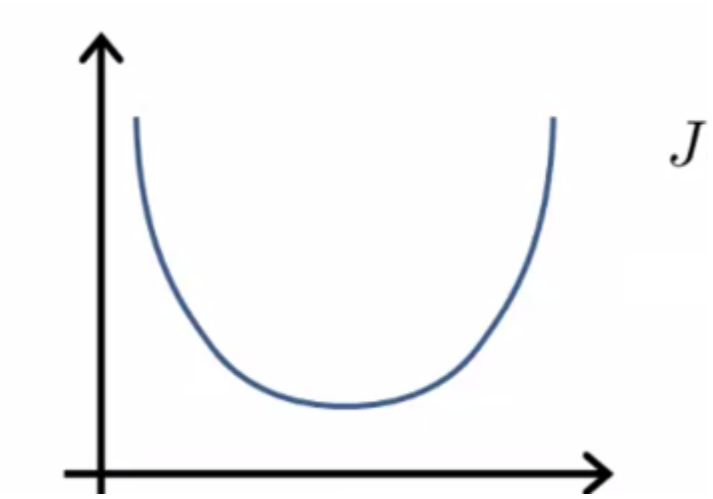
With vectorization

$$\vec{w} = \vec{w} - 0.1\vec{d}$$

```
w = w - 0.1 * d
```

# Linear Regression – Normal Equation

Gradient Descent



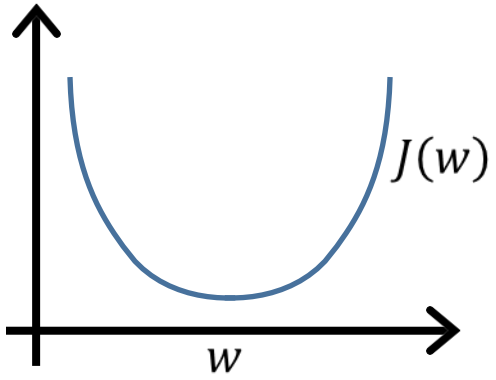

Normal equation: Method to solve for $\mathbf{w}$ analytically.

# Linear Regression – Normal Equation (Cont.)

Intuition: If 1D $w \in \mathbb{R}$

$$J(w) = aw^2 + bw + c$$

$$\frac{\partial}{\partial w_j} J(\vec{w}) = \ldots = 0$$



# Linear Regression – Normal Equation (Cont.)

Examples: $m = 4$.

| $x_0$ | Size (feet²) $x_1$ | Number of bedrooms $x_2$ | Number of floors $x_3$ | Age of home (years) $x_4$ | Price ($1000) $y$ |
|---|---|---|---|---|---|
| 1 | 2104 | 5 | 1 | 45 | 460 |
| 1 | 1416 | 3 | 2 | 40 | 232 |
| 1 | 1534 | 3 | 2 | 30 | 315 |
| 1 | 852 | 2 | 1 | 36 | 178 |

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix} \qquad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}$$

$$\mathbf{w} = (X^T X)^{-1} X^T y$$

# Linear Regression – Normal Equation (Cont.)

$m$ **examples** $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$ **;** $n$ **features.**

$$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^{n+1}$$

E.g. If $x^{(i)} = \begin{bmatrix} 1 \\ x_1^{(i)} \end{bmatrix}$

**w** = $(X^T X)^{-1} X^T y$

```
pinv(X'*X)*X'*y
```

$(X^T X)^{-1}$ is inverse of matrix $X^T X$

# Linear Regression – Normal Equation (Cont.)

**training examples, $n$ features.**

| Gradient Descent | Normal Equation |
| --- | --- |
| • Need to choose $\alpha$.<br>• Needs many iterations.<br>• Works well eve when $n$ is large. | • No need to choose $\alpha$.<br>• Don't need to iterate.<br>• Need to compute $(X^T X)^{-1}$<br>• Slow if $n$ is very large. |

# Derivation of the Normal Equation

1. **Cost Function**
   - **Summation Form** (with $m$ samples, $n+1$ parameters):

$$J(\mathbf{w}) = \frac{1}{2m} \sum_{i=1}^{m} \left(\mathbf{w}^\top \mathbf{x}^{(i)} - y^{(i)}\right)^2.$$

   - **Matrix Form**:

$$J(\mathbf{w}) = \frac{1}{2m} \left(X\mathbf{w} - \mathbf{y}\right)^\top \left(X\mathbf{w} - \mathbf{y}\right).$$

   Here,

   - $X$ is $m \times (n+1)$,
   - $\mathbf{w}$ is $(n+1) \times 1$,
   - $\mathbf{y}$ is $m \times 1$.

# Derivation of the Normal Equation (cont.)

2. **Expand** $(X\mathbf{w} - \mathbf{y})^\top (X\mathbf{w} - \mathbf{y})$

$$(X\mathbf{w} - \mathbf{y})^\top (X\mathbf{w} - \mathbf{y}) = (X\mathbf{w})^\top (X\mathbf{w}) - (X\mathbf{w})^\top \mathbf{y} - \mathbf{y}^\top (X\mathbf{w}) + \mathbf{y}^\top \mathbf{y}.$$

- Rewrite each term:

$$(X\mathbf{w})^\top (X\mathbf{w}) = \mathbf{w}^\top X^\top X\mathbf{w}, \quad (X\mathbf{w})^\top \mathbf{y} = \mathbf{w}^\top X^\top \mathbf{y},$$

  and note $\mathbf{w}^\top X^\top \mathbf{y} = \mathbf{y}^\top X\mathbf{w}$ (a scalar).

- Therefore,

$$(X\mathbf{w} - \mathbf{y})^\top (X\mathbf{w} - \mathbf{y}) = \mathbf{w}^\top X^\top X\mathbf{w} - 2\,\mathbf{w}^\top X^\top \mathbf{y} + \mathbf{y}^\top \mathbf{y}.$$

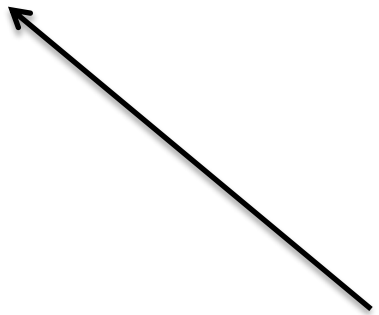
- $X$ is $m \times (n+1)$,
- $\mathbf{w}$ is $(n+1) \times 1$,
- $\mathbf{y}$ is $m \times 1$.

1. $(A^T)^T = A$
2. $(A+B)^T = A^T + B^T$ *and* $(A-B)^T = A^T - B^T$
3. $(kA)^T = kA^T$
4. $(AB)^T = B^T A^T$

# Derivation of the Normal Equation (cont.)

Because $\mathbf{y}^\top \mathbf{y}$ does not depend on $\mathbf{w}$, its gradient is zero. The other two pieces yield the well-known results from matrix calculus:

1. $\nabla_{\mathbf{w}}\left(\mathbf{w}^\top X^\top X \mathbf{w}\right) = 2\, X^\top X\, \mathbf{w}.$
2. $\nabla_{\mathbf{w}}\left(\mathbf{w}^\top X^\top \mathbf{y}\right) = X^\top \mathbf{y}.$



3. **Take the Gradient**

$$J(\mathbf{w}) = \frac{1}{2m}\left(\mathbf{w}^\top X^\top X \mathbf{w} - 2\,\mathbf{w}^\top X^\top \mathbf{y} + \mathbf{y}^\top \mathbf{y}\right).$$

- Differentiate w.r.t. $\mathbf{w}$:

$$\nabla_{\mathbf{w}}\, J(\mathbf{w}) = \frac{1}{2m}\,\nabla_{\mathbf{w}}\left(\mathbf{w}^\top X^\top X \mathbf{w} - 2\,\mathbf{w}^\top X^\top \mathbf{y}\right) = \frac{1}{2m}\left(2\, X^\top X\, \mathbf{w} \;-\; 2\, X^\top \mathbf{y}\right).$$

- Simplify:

$$\nabla_{\mathbf{w}}\, J(\mathbf{w}) = \frac{1}{m}\left(X^\top X\, \mathbf{w} - X^\top \mathbf{y}\right).$$

# Derivation of the Normal Equation (cont.)

## 4. Set the Gradient to Zero and Solve

For a **minimum**, set the gradient to the **zero** vector:

$$\frac{1}{m}\left(X^\top X\,\mathbf{w} \;-\; X^\top \mathbf{y}\right) = \mathbf{0}.$$

Multiply both sides by $m$ (which changes nothing) and rearrange:

$$X^\top X\,\mathbf{w} = X^\top \mathbf{y}.$$

If $(X^\top X)$ is **invertible**, multiply by its inverse:

$$\mathbf{w} = \left(X^\top X\right)^{-1} X^\top \mathbf{y}.$$